

DESIGN AND IMPLEMENTATION OF FPGA BASED LOW POWER 32- BIT BARREL SHIFTER

Gudapati Lakshmi Bhavani¹, M.Nithin², T.Lokesh^{3*}V Sheshank⁴

^{1,2,3,4}KG REDDY College Of Engineering & Technology, Hyderabad,Telangana-India

⁴Dept of Electronics and Communication Engineering

Corresponding Author *: lakshmisrinicas.2010@kgr.ac.in

FPGA ఆధారిత తక్కువ శక్తి 32- బిట్ బారెల్ షిఫ్టర్ రూపకల్పన మరియు అమలు

గూడపాటి లక్ష్మి భవాని¹, ఎం. నితీన్², టి.లోకేశ్^{3*}వి శేషాంక్⁴
^{1,2,3,4} కేజీ రెడ్డి కాలేజ్ ఆఫ్ ఇంజనీరింగ్ & టెక్నాలజీ, హైదరాబాద్, తెలంగాణ-భారతదేశం
⁴ ఎలక్ట్రానిక్స్ మరియు కమ్యూనికేషన్ ఇంజనీరింగ్ విభాగం
సంబంధిత రచయిత *: lakshmisrinicas.2010@kgr.ac.in

సారాంశం- ఈ ప్రాజెక్ట్ FPGA లో తక్కువ-శక్తి, అధిక-పనితీరు గల 32-బిట్ బారెల్ షిఫ్టర్ యొక్క రూపకల్పన మరియు అమలును అందిస్తుంది. బారెల్ షిఫ్టర్ అనేది బహుళ చక్రాలు అవసరమయ్యే సాంప్రదాయ సీక్వెన్షియల్ షిఫ్టర్ల మాదిరిగా కాకుండా, ఒకే చక్రంలో వేగంగా బిట్ వైజ్ షిఫ్టింగ్ మరియు భ్రమణాన్ని నిర్వహించడానికి ఆధునిక ప్రాసెసర్లలో విస్తృతంగా ఉపయోగించే కాంబినేషన్ సర్క్యూట్. వెరిలాగ్ HDLలో అభివృద్ధి చేయబడిన డిజైన్, Xilinx వంటి FPGA ప్లాట్ ఫారమ్లపై అనుకరించబడింది మరియు సంశ్లేషణ చేయబడింది. ఇది నాలుగు ఆపరేషన్లకు మద్దతు ఇస్తుంది: లాజికల్ లెఫ్ట్ షిఫ్ట్ (LSL), లాజికల్ రైట్ షిఫ్ట్ (LSR), అంకగణిత రైట్ షిఫ్ట్ (ASR), మరియు రొటేట్ రైట్ (ROR). ఆలస్యాన్ని తగ్గించడానికి మరియు హార్డ్వేర్ వినియోగాన్ని ఆప్టిమైజ్ చేయడానికి క్రమానుగత మల్టీప్లెక్సర్ నిర్మాణం ఉపయోగించబడుతుంది. పోస్ట్-సింథసిస్ ఫలితాలు సరైన కార్యాచరణ, సమర్థవంతమైన పనితీరు మరియు తక్కువ వనరుల వినియోగాన్ని నిర్ధారిస్తాయి, లుక్-అప్ టేబుల్స్ (LUTలు) మరియు ఇన్ పుట్/అవుట్ పుట్ బ్లాక్స్ (IOBలు) యొక్క చిన్న భాగాన్ని మాత్రమే వినియోగిస్తాయి. పవర్ విశ్లేషణ కనిష్ట స్టాటిక్ మరియు డైనమిక్ పవర్ వినియోగాన్ని చూపుతుంది, చాలా డైనమిక్ పవర్ I/O కార్యాచరణకు ఆపాదించబడింది, ఇది ఎంబెడెడ్ మరియు పోర్టబుల్ సిస్టమ్లకు బాగా సరిపోతుంది. మొత్తంమీద, డిజైన్ ఆధునిక డిజిటల్ అప్లికేషన్లలో విస్తరణకు సిద్ధంగా ఉన్న స్కేలబుల్, తక్కువ-శక్తి మరియు బలమైన బారెల్ షిఫ్టర్ నిర్మాణాన్ని ప్రదర్శిస్తుంది.

కీలకపదాలు:

బారెల్ షిఫ్టర్, FPGA ఇంప్లిమెంటేషన్, లో-పవర్ డిజైన్, హై-పెర్ఫార్మెన్స్ ఆర్కిటెక్చర్, వెరిలాగ్ HDL, లాజికల్ షిఫ్ట్ లెఫ్ట్ (LSL), లాజికల్ షిఫ్ట్ రైట్ (LSR), అరిథ్మెటిక్ షిఫ్ట్ రైట్ (ASR), రొటేట్ రైట్ (ROR), మల్టీప్లెక్సర్-బేస్డ్ డిజైన్, రిసోర్స్ యుటిలైజేషన్.

1. పరిచయం

బారెల్ షిఫ్టర్ అనేది ఒక కాంబినేషన్ సర్క్యూట్, ఇది ఒకే చక్రంలో వేగంగా బిట్వైజ్ షిఫ్టింగ్ లేదా రొటేషన్‌ను నిర్వహిస్తుంది, ఇది ప్రాసెసర్లు, DSP, క్రిప్టోగ్రఫీ మరియు ఎంబెడెడ్ సిస్టమ్‌లలో విస్తృతంగా ఉపయోగించబడుతుంది. సాంప్రదాయ షిఫ్ట్ రిజిస్టర్‌ల మాదిరిగా కాకుండా, ఇది సమర్థవంతమైన ఆపరేషన్‌ను సాధించడానికి మల్టీప్లెక్సర్‌లను (MUX) ఉపయోగించి బహుళ-దశల ఎంపికను ఉపయోగిస్తుంది. గుణకారం, విభజన, ప్లోటింగ్-పాయింట్ సాధారణీకరణ మరియు బిట్ మానిప్యులేషన్ కోసం షిఫ్టింగ్ చాలా ముఖ్యమైనది. FPGA లు వాటి సమాంతరత మరియు హై-స్పీడ్ ప్రాసెసింగ్ కారణంగా బారెల్ షిఫ్టర్‌లను అమలు చేయడానికి ఆదర్శవంతమైన వేదికను అందిస్తాయి. 32-బిట్ బారెల్ షిఫ్టర్‌ను 1, 2, 4, 8 మరియు 16-బిట్ దశలతో క్రమానుగత నిర్మాణాన్ని ఉపయోగించి రూపొందించవచ్చు, ప్రచార ఆలస్యాన్ని తగ్గిస్తుంది మరియు వనరుల వినియోగాన్ని ఆప్టిమైజ్ చేస్తుంది. డిజైన్ వెరిలాగ్ లేదా VHDL ఉపయోగించి అమలు చేయబడుతుంది మరియు Xilinx Vivado/ISE లేదా Intel Quartus వంటి FPGA ప్లాట్‌ఫారమ్‌లలో పరిక్షించబడుతుంది. ఈ ప్రాజెక్ట్ సాఫ్ట్‌వేర్ పద్ధతులపై హార్డ్‌వేర్ ఆధారిత షిఫ్టింగ్ యొక్క ప్రయోజనాలను ప్రదర్శిస్తుంది, వేగం, సామర్థ్యం మరియు సమాంతరతలో మెరుగుదలలను హైలైట్ చేస్తుంది. ఎంబెడెడ్ సిస్టమ్స్, క్రిప్టోగ్రఫీ మరియు మెషిన్ లెర్నింగ్ యాక్సిలరేటర్లు వంటి ఆధునిక అనువర్తనాల్లో బారెల్ షిఫ్టర్ ప్రాముఖ్యతను అమలు మరియు నొక్కి చెబుతుంది. . పరిశోధన లక్ష్యాలు:

1. FPGA ప్లాట్‌ఫామ్‌పై వెరిలాగ్ HDLని ఉపయోగించి తక్కువ-శక్తి, అధిక-పనితీరు గల 32-బిట్ బారెల్ షిఫ్టర్‌ను రూపొందించడానికి మరియు అమలు చేయడానికి, LSL, LSR, ASR మరియు ROR వంటి బహుళ షిఫ్ట్ మరియు రొటేట్ ఆపరేషన్‌లకు మద్దతును నిర్ధారిస్తుంది.
2. ప్రచార ఆలస్యాన్ని తగ్గించడం, హార్డ్‌వేర్ వనరుల వినియోగాన్ని తగ్గించడం మరియు పెద్ద డిజిటల్ సిస్టమ్‌లలో ఏకీకరణ కోసం స్కేలబిలిటీని సాధించడం అనే లక్ష్యంతో, క్రమానుగత మల్టీప్లెక్సర్-ఆధారిత నిర్మాణాన్ని ఉపయోగించి బారెల్ షిఫ్టర్ ఆర్కిటెక్చర్‌ను ఆప్టిమైజ్ చేయడం.

3. FPGA సంశ్లేషణ మరియు అమలు తర్వాత విశ్లేషణ ద్వారా ప్రతిపాదిత బారెల్ షిఫ్టర్ యొక్క క్రియాత్మక ఖచ్చితత్వం, వనరుల సామర్థ్యం మరియు విద్యుత్ వినియోగాన్ని అంచనా వేయడం, ఎంబెడెడ్ మరియు పోర్టబుల్ తక్కువ-శక్తి అనువర్తనాలకు దాని అనుకూలతను ధృవీకరించడం.

2. సాహిత్య సర్వే

ఇటీవలి పరిశోధనలు విభిన్న అప్లికేషన్లలో పనితీరును మెరుగుపరచడం, శక్తిని తగ్గించడం మరియు స్కేలబిలిటీని మెరుగుపరచడంపై దృష్టి సారించాయి. పాపగేర్గియో (2025) హై-త్రూపుట్ సిగ్నల్ మరియు ఇమేజ్ ప్రాసెసింగ్ కోసం నవల షెడ్యూలింగ్ మరియు షిఫ్టర్ నెట్వర్క్ ఆర్కిటెక్చర్లను ప్రవేశపెట్టారు. శర్మ, పురోహిత మరియు అసతి (2025) 5G సిస్టమ్ల కోసం FPGA లో సింగిల్-సైకిల్ LDPC ఎన్కోడర్ను అమలు చేశారు, అయితే లి మరియు ఇతరులు (2025) ఖచ్చితత్వం మరియు తక్కువ శక్తి కోసం స్వీయ-క్యాలిబ్రేషన్తో 14-బిట్ హైబ్రిడ్ SAR ADCని ప్రతిపాదించారు. క్రిష్ణోగ్రాఫిక్ హార్డ్వేర్లో, లీ మరియు ఇతరులు (2025) GIFT-128 సైఫర్ కోసం ఇంటర్లీవ్ పరిమాణాలను అధ్యయనం చేశారు మరియు రాహుల్ మరియు ఇతరులు (2025) శక్తి మరియు ప్రాంతాన్ని ఆదా చేయడానికి వేద గణితాన్ని ఉపయోగించి RISC కోర్ను రూపొందించారు. అదేవిధంగా, ఇజాది మరియు జంషిది (2025) తక్కువ-శక్తి ఉజ్జాయింపు గుణకాన్ని అందించాడు మరియు సుమారుదీన్ మరియు ఇతరులు (2025) ఎడ్జ్ కంప్యూటింగ్లో లోతైన Q-నెట్వర్క్ల కోసం డ్యూయల్-మోడ్ యాక్సిలరేటర్ను అభివృద్ధి చేశారు. జాయిస్ మరియు ఇతరులు (2025) రద్దీగా ఉండే డేటాపాత్లలో పవర్ గ్రిడ్లను ఆప్టిమైజ్ చేయడంతో, అగర్వాల్, గార్గ్ మరియు కుమార్ (2025) FPGA-ఆధారిత తక్కువ-శక్తి వ్యవస్థలను సమర్థించడంతో, ములే మరియు మండే (2025) CORDIC మాడ్యూల్స్ ద్వారా FFT సంక్లిష్టతను తగ్గించడంతో, మోడియా మరియు ఇతరులు (2025) వేద గుణకాన్ని అమలు చేయడంతో, ఎర్నా మరియు ఇతరులు (2025) అధిక-త్రూపుట్ హాఫ్మాన్ కోడెక్ను గ్రహించడంతో విద్యుత్-సమర్థవంతమైన డిజైన్ కూడా దృష్టిని ఆకర్షించింది. బారెల్ షిఫ్టర్లకు ప్రత్యేకంగా, దేవమనే మరియు ఇతరులు (2015) ఫంక్షనల్ కరెక్ట్నెస్ మరియు రిసోర్స్ సామర్థ్యాన్ని నొక్కిచెప్పగా, రెడ్డి మరియు బాబు (2016) హై-స్పీడ్ డేటా ప్రాసెసింగ్ కోసం 32-బిట్ FPGA-ఆధారిత బారెల్ షిఫ్టర్ల అనుకూలతను హైలైట్ చేశారు. ఈ పనులు సమిష్టిగా ఆధునిక ఎంబెడెడ్ మరియు పోర్టబుల్ సిస్టమ్ల కోసం స్కేలబుల్, తక్కువ-శక్తి మరియు అధిక-పనితీరు గల బారెల్ షిఫ్టర్ ఆర్కిటెక్చర్లపై ప్రస్తుత పరిశోధనకు పునాదిని అందిస్తాయి.

3. పద్ధతి

లక్ష్యాలను సాధించడానికి 32-బిట్ బారెల్ షిఫ్టర్ యొక్క ఫ్లోచార్ట్ మొత్తం డిజైన్ మరియు అమలు ప్రక్రియలో తీసుకున్న క్రమబద్ధమైన దశలను సూచిస్తుంది. ఇది బారెల్ షిఫ్టర్ యొక్క అవసరాలను అర్థం

చేసుకోవడంతో ప్రారంభమవుతుంది, దాని ఇన్పుట్/అవుట్పుట్ నిర్మాణం మరియు అది నిర్వహించాల్సిన ఆపరేషన్ల రకాలు. సమస్యను నిర్వచించిన తర్వాత, తదుపరి దశ వెరిలాగ్ HDLని ఉపయోగించి లాజిక్ ను రూపొందించడం .



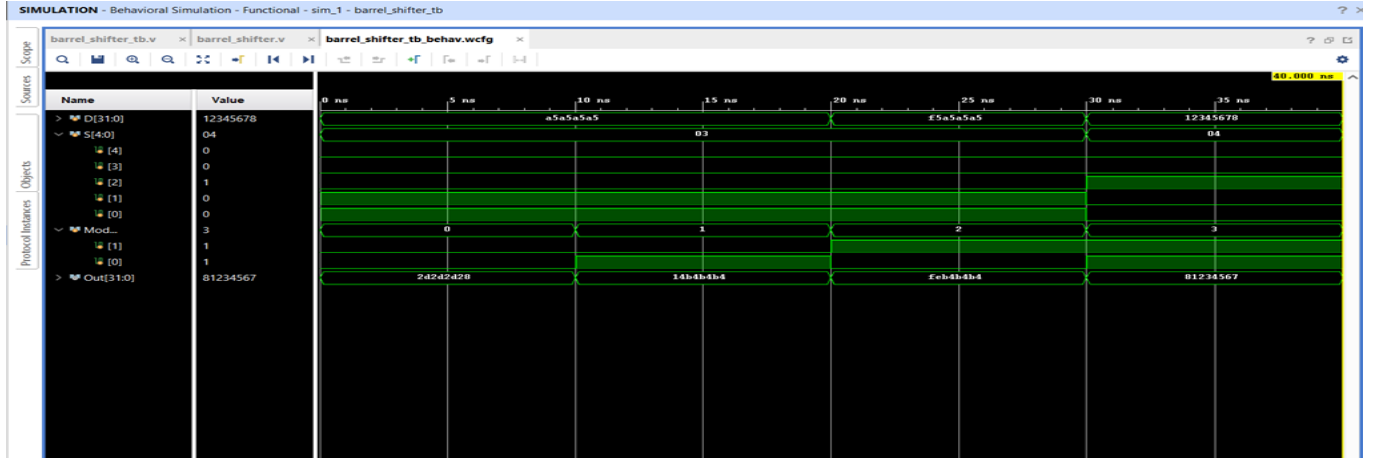
చిత్రం1: 32-బిట్ బారెల్ షిఫ్టర్ యొక్క ప్లోచార్ట్ .

32-బిట్ బారెల్ షిఫ్టర్ యొక్క ప్లోచార్ట్ మొత్తం డిజైన్ మరియు అమలు ప్రక్రియలో తీసుకున్న క్రమబద్ధమైన దశలను సూచిస్తుంది. ఇది బారెల్ షిఫ్టర్ యొక్క అవసరాలను అర్థం చేసుకోవడంతో ప్రారంభమవుతుంది, దాని ఇన్పుట్/అవుట్పుట్ నిర్మాణం మరియు అది నిర్వహించాల్సిన ఆపరేషన్ల రకాలు ఉన్నాయి. సమస్యను నిర్వచించిన తర్వాత, తదుపరి దశ వెరిలాగ్ HDLని ఉపయోగించి లాజిక్ ను రూపొందించడం, నియంత్రణ సంకేతాల ఆధారంగా డేటా ఎలా మార్చబడుతుందో లేదా తిప్పబడుతుందో పేర్కొనడం. [3]

4. ఫలితాల విశ్లేషణ

ప్రవర్తనా అనుకరణ ఫలితంలో చూపబడిన తరంగ రూపం వెరిలాగ్ ను ఉపయోగించి 32-బిట్ బారెల్ షిఫ్టర్ యొక్క క్రియాత్మక ధృవీకరణను సూచిస్తుంది. కాలక్రమేణా వివిధ షిఫ్ట్ మోడ్లు మరియు ఇన్పుట్ పరిస్థితుల కోసం బారెల్ షిఫ్టర్ యొక్క అవుట్పుట్ ప్రతిస్పందనను గమనించడానికి అనుకరణ నిర్వహించబడుతుంది. ఇన్పుట్ 0[31:0] మార్చవలసిన 32-బిట్ డేటాను కలిగి ఉంటుంది మరియు 5[4:0] షిఫ్ట్ మొత్తాన్ని నిర్దేశిస్తుంది. నియంత్రణ సిగ్నల్ మోడ్[1:0] షిఫ్ట్ ఆపరేషన్ రకాన్ని నిర్ణయిస్తుంది: లాజిక్ లెఫ్ట్ షిఫ్ట్ (00), లాజిక్ రైట్ షిఫ్ట్ (01), అంకగణిత కుడి షిఫ్ట్ (10), లేదా కుడివైపు తిప్పండి (11). తరంగ రూపంలో, వేర్వేరు సమయ విరామాలు (నానోసెకన్లలో) ఈ నియంత్రణ సిగ్నల్లు మరియు ఇన్పుట్ డేటాలో ప్రవర్తనా అనుకరణ మార్పులను ప్రదర్శిస్తాయి, అవుట్పుట్ అవుట్పుట్[31:0]లో సంబంధిత మార్పులు గమనించబడతాయి. ఉదాహరణకు, ఒక సమయంలో, మోడ్ "11" (కుడివైపుకు

తిప్పండి)కి సెట్ చేయబడినప్పుడు మరియు షిఫ్ట్ మొత్తం "4" అయినప్పుడు, అవుట్పుట్ బిట్ల వృత్తాకార భ్రమణాన్ని ప్రతిబింబిస్తుంది. బారెల్ షిఫ్టర్ మోడ్ మరియు షిఫ్ట్ ఇన్పుట్లను సరిగ్గా అర్థం చేసుకుంటుందని, నిజ సమయంలో అంచనా వేసిన అవుట్పుట్ విలువలను ఉత్పత్తి చేస్తుందని అనుకరణ నిర్ధారిస్తుంది. ఇది వాటి పనితీరు కొలమానాల ఆధారంగా వివిధ నమూనాలను నిర్వహించేటప్పుడు డిజైన్ యొక్క కార్యాచరణను ధృవీకరిస్తుంది మరియు ఫలితాలను విశ్లేషిస్తుంది.



చిత్రం 2: దీని కోసం అనుకరణ వెరిలాగ్ ఉపయోగించి ప్రవర్తనా అనుకరణ 32-బిట్ బారెల్ షిఫ్టర్.

RTL (రిజిస్టర్ ట్రాన్స్ఫర్ లెవల్) స్క్రిమాటిక్ అనేది వివరణ తర్వాత ఉత్పత్తి చేయబడిన HDL కోడ్ యొక్క నిర్మాణాత్మక ప్రాతినిధ్యం, ఇది డిజైన్ రిజిస్టర్లు, మల్టీప్లెక్సర్లు, గేట్లు మరియు నియంత్రణ మార్గాలలో తార్కికంగా ఎలా నిర్వహించబడిందో చూపిస్తుంది. 32-బిట్ బారెల్ షిఫ్టర్ కోసం, RTL స్క్రిమాటిక్ వెరిలాగ్ లేదా VHDL నుండి స్వయంచాలకంగా ఉత్పత్తి చేయబడిన డేటా ప్రవాహం యొక్క అధిక-స్థాయి సంగ్రహణను అందిస్తుంది. ఇందులో 32-బిట్ డేటా లైన్ (D[31:0]), 5-బిట్ షిఫ్ట్ కంట్రోల్ (S[4:0]) మరియు 2-బిట్ మోడ్ సెలెక్టర్ (మోడ్[1:0]) వంటి ఇన్పుట్ పోర్ట్లు ఉంటాయి. అంతర్గతంగా, డిజైన్ 1, 2, 4, 8 మరియు 16 బిట్ల దశల్లో డేటాను మార్చే మల్టీప్లెక్సర్ల క్రమానుగత అమరికను ఉపయోగిస్తుంది, ఇది వేగవంతమైన సమాంతర బదిలీని అనుమతిస్తుంది. మోడ్[1:0] ఆధారంగా నియంత్రణ లాజిక్, అవసరమైన ఆపరేషన్‌ను సక్రియం చేస్తుంది—లాజిక్ షిఫ్ట్ (LSL), లాజిక్ రైట్ షిఫ్ట్ (LSR), అరిథ్మెటిక్ రైట్ షిఫ్ట్ (ASR), లేదా రొటేట్ రైట్ (ROR)—మరియు ఫలితం 32-బిట్ అవుట్ పుట్కు పంపబడుతుంది (అవుట్[31:0]). RTL స్క్రిమాటిక్ ఫంక్షనల్ కరెక్టెనెస్ను ధృవీకరించడంలో, కనెక్ట్ చేయని పోర్ట్లు లేదా రిడండ్ లాజిక్ వంటి సమస్యలను గుర్తించడంలో మరియు ఉద్దేశించిన

ఆర్కిటెక్చర్ను డాక్యుమెంట్ చేయడంలో సహాయపడుతుంది. దీనికి విరుద్ధంగా, పోస్ట్-సింథసిస్ స్క్రిమాటిక్ Xilinx Vivado, Synopsys Design Compiler లేదా Intel Quartus వంటి సాధనాలలో సంశ్లేషణ తర్వాత అదే డిజైన్ యొక్క గేట్-స్థాయి ప్రాతినిధ్యాన్ని అందిస్తుంది. ఈ స్క్రిమాటిక్ FPGA వనరులకు మ్యాప్ చేయబడిన ఆప్టిమైజ్డ్ చేయబడిన హార్డ్వేర్ అమలును ప్రతిబింబిస్తుంది, హై-లెవల్ RTL వివరణ వైశాల్యం, వేగం మరియు శక్తి యొక్క పరిమితుల కింద ప్రామాణిక లాజిక్ గేట్లు మరియు ఇంటర్కనెక్షన్ లలో ఎలా అనువదించబడుతుందో చూపిస్తుంది. 32-బిట్ బారెల్ షిఫ్టర్ కోసం, ఇది డిజైన్ విస్తరణకు ముందు క్రియాత్మకంగా సరైనదని, వనరు-సమర్థవంతమైనదని మరియు పనితీరు-ఆప్టిమైజ్డ్ చేయబడిందని నిర్ధారిస్తుంది.

పట్టిక 1: నివేదికలు ఆఫ్ 8,16,32, బిట్స్ ఆఫ్ బారెల్ షిఫ్టర్

లాజిక్(8) బిట్)	ఉపయోగించబడింది	అందుబాటులో ఉంది	యుటిలైజేషన్
లేదు. ముక్కలు	18	860 తెలుగు in లో	1%
లేదు. యొక్క LUT లు	33	1540 తెలుగు in లో	1%
లేదు. యొక్క IOB లు	16	46 తెలుగు	22%
లాజిక్(16) బిట్)	ఉపయోగించబడింది	అందుబాటులో ఉంది	యుటిలిజేషన్
లేదు. ముక్కలు	55	910 తెలుగు in లో	4%
లేదు. యొక్క LUT లు	101 తెలుగు	1590 తెలుగు in లో	4%
లేదు. యొక్క IOB లు	22	56 తెలుగు	46%
లాజిక్(32) బిట్)	ఉపయోగించబడింది	అందుబాటులో ఉంది	యుటిలిజేషన్
లేదు. యొక్క LUT లు	230 తెలుగు in లో	8000 నుండి 8000 వరకు	3%
లేదు. యొక్క IOB లు	71 అనుక్షిప్త	100 లు	71%

5. ముగింపు

32-బిట్ తక్కువ-శక్తి గల బారెల్ షిఫ్టర్ విజయవంతంగా రూపొందించబడింది మరియు అమలు చేయబడింది, ఇది లాజిక్, అంకగణిత మరియు భ్రమణ షిఫ్ట్ ఆపరేషన్లకు మద్దతు ఇస్తుంది. వివిధ నియంత్రణ సంకేతాల కింద డిజైన్ సరైన మరియు ఊహించదగిన అవుట్పుట్లను ఉత్పత్తి చేసిందని

ఫంక్షనల్ సిమ్యులేషన్ ధృవీకరించింది. సంశ్లేషణ ఫలితాలు LUT లు మరియు I/O పిన్ల కనీస వినియోగాన్ని చూపించాయి, డిజైన్ కాంపాక్ట్గా మరియు పెద్ద డిజిటల్ సిస్టమ్లలో సులభంగా సమగ్రపరచదగినదిగా చేసింది. పవర్ విశ్లేషణ తక్కువ డైనమిక్ మరియు స్టాటిక్ పవర్ వినియోగాన్ని నిర్ధారించింది, అధిక-బిట్-వెడల్పు ఆపరేషన్లలో I/O కార్యాచరణకు చాలా డైనమిక్ పవర్ ఆపాదించబడింది. లాజిక్ మూలకాల యొక్క సమతుల్య పంపిణీ రూటింగ్ ఆలస్యాన్ని తగ్గించడానికి మరియు మొత్తం వేగాన్ని మెరుగుపరచడంలో సహాయపడింది. ఇంకా, డిజైన్ 8-బిట్, 16-బిట్ మరియు 32-బిట్ వెర్షన్లలో కనీస ఓవర్హెడ్తో స్కేలబిలిటీని ప్రదర్శించింది. మొత్తంమీద, ప్రతిపాదిత బారెల్ షిఫ్టర్ పవర్, పనితీరు మరియు వైశాల్యం యొక్క ముఖ్యమైన పరిమితులను తీరుస్తుంది, ఇది ఆధునిక VLSI సిస్టమ్లు మరియు ఎంబెడెడ్ అప్లికేషన్లకు అత్యంత అనుకూలంగా ఉంటుంది.

6. సూచనలు

పాపగేర్రియో, ఎన్. (2025). నవల షెడ్యూలింగ్ మరియు షిఫ్టర్ నెట్వర్క్లు. సిగ్నల్ మరియు ఇమేజ్ ప్రాసెసింగ్ కోసం డిజైన్ మరియు ఆర్కిటెక్చర్లో: 18వ అంతర్జాతీయ వర్క్ షాప్, DASIP 2025, బార్నిలోసా, స్పెయిన్, జనవరి 20-22, 2025, ప్రొసీడింగ్స్ (వాల్వ్యూమ్. 15569, పేజీ. 95). స్ప్రింగర్ నేచర్.

శర్మ, ఎ., పురోహిత్, జి., & అసతి, ఎఆర్ (2025). 5G కొత్త రేడియో కోసం సింగిల్-సైకిల్ హై-త్రాపుట్ LDPC ఎన్కోడర్ యొక్క FPGA -ఆధారిత అమలు. IEEE ఎంబెడెడ్ సిస్టమ్స్ లెటర్స్.

లి, ఎస్., మెంగ్, క్యూ., జాంగ్, ఎల్., & వు, జె. (2025). కామన్-మోడ్ స్వీయ-కాలిబ్రేషన్తో కూడిన 14-బిట్ 2-MS/s హైబ్రిడ్-లాజిక్ ఆధారిత SAR ADC. మైక్రోఎలక్ట్రానిక్స్ జర్నల్, 1067.

లీ, ఎం., షిన్, హెచ్., షిన్, ఎం., లీ, డి., సియో, ఎస్సి, & హాంగ్, ఎస్. (2025). 64-బిట్ ARM ప్రాసెసర్లో GI FT -128 యొక్క ఉత్తమ ఇంటర్లీవ్ సైజు. జర్నల్ ఆఫ్ క్రిస్టోగ్రాఫిక్ ఇంజనీరింగ్, 15 (1), 1-16.

రాహుల్, డి., లక్ష్మి, జి.పి., రెడ్డి, జి.ఎం., & గనోర్కర్, ఎ.ఎం. (2025). వేద గణితాన్ని ఉపయోగించి తక్కువ శక్తి మరియు తక్కువ వైశాల్యం గల RI SC కోర్ రూపకల్పన మరియు అమలు. 2025లో IEEE ఇంటర్నేషనల్ కాన్ఫరెన్స్ ఆన్ ఇంటర్ డిస్ట్రినరీ అప్రోచెస్ ఇన్ టెక్నాలజీ అండ్ మేనేజ్మెంట్ ఫర్ సోషల్ ఇన్నోవేషన్ (IATMSI) (వాల్వ్యూమ్. 3, పేజీలు. 1-6). IEEE.

ఇజాది, ఎ., & జంషిది, వి. (2025). LHTAM: చిన్న ఖచ్చితమైన కంప్యూటింగ్ వ్యవస్థల కోసం తక్కువ-శక్తి మరియు అధిక-వేగ ఉజ్జాయింపు గుణకం. కంప్యూటర్లు & ఎలక్ట్రోనిక్ ఇంజనీరింగ్, 123, 110215.

సుమారుదీన్, ఎ., సుటిస్సా, ఎన్., సయాఫల్ని, ఐ., ట్రిలాక్సోనో, బిఆర్, & అడియోనో, టి. (2025). డీప్ క్యూ-నెట్వర్క్ (DQN) ఆధారిత ఎడ్జ్ కంప్యూటింగ్ కోసం డ్యూయల్-మోడ్ శిక్షణ మరియు అనుమితి హార్డ్వేర్ యాక్సిలరేటర్ యొక్క తక్కువ సంక్లిష్టత మరియు స్కేలబుల్ ఆర్కిటెక్చర్. *IEEE యాక్సెస్*.

జాయిస్, సిఎం, ఉపాధ్యాయ, పి., నిషాద్, ఎస్., & కక్కర్, ఎ. (2025). రద్దీగా ఉండే హై ఫ్రీక్వెన్సీ డేటాపాత్ డిజైన్ల కోసం పిపిఎ-అవేర్ పవర్ గ్రిడ్ ఆప్టిమైజేషన్ టెక్నిక్స్. *2025లో 38వ అంతర్జాతీయ VLSI డిజైన్ కాన్ఫరెన్స్ మరియు 2024లో 23వ అంతర్జాతీయ ఎంబెడెడ్ సిస్టమ్స్ కాన్ఫరెన్స్ (VLSI D) (పేజీలు 308-313)*. IEEE.

అగర్వాల్, పి., గార్గ్, టి, & కుమార్, ఎ. (2025). FPGA లను ఉపయోగించి తక్కువ-శక్తి ఎంబెడెడ్ సిస్టమ్ డిజైన్ అప్లికేషన్లు. *ఎంబెడెడ్ డివైజెస్ మరియు ఇంటర్నెట్ ఆఫ్ థింగ్స్ (పేజీలు 1-20)*. CRC ప్రెస్.

మ్యూల్, PC , & మండే, SS (2025). ఎంబెడెడ్ CORDIC మాడ్యూల్తో FFT ట్వీడిల్ ఫ్యాక్టర్ బటర్ఫ్లె యొక్క సంక్లిష్టతలో FPGA -ఆధారిత తగ్గింపు. *ఇంటర్నేషనల్ జర్నల్ ఆఫ్ అడ్ హాక్ అండ్ యుబిక్విటస్ కంప్యూటింగ్, 48 (1)*, 34-45.

మోడియా, హెచ్., చౌహాన్, డి., పటేల్, ఎస్., గొండాలియా, వి., & ఖస్తారియా, జె. (2025). వెరిలాగ్ HDL ఉపయోగించి స్పార్ట్-6 FPGA పై వేదిక్ గుణకం యొక్క రూపకల్పన మరియు అమలు. *[ప్రచురణ వివరాలు లేవు]*.

ఎర్నా, జి., సైదులు, వి., కుడితి, టి., & బానోత్, ఎఎన్ (2025). లాస్లెస్ కానానికల్ హాఫ్మన్ యంత్రం యొక్క అధిక త్రూపుట్ ఎన్కోడర్ మరియు డీకోడర్ డిజైన్ యొక్క FPGA అమలు. *ఇంజనీరింగ్లో ఫలితాలు, 105037*.

దేవమనే, ఎస్., హంచటి, ఎ., వాగరే, యు., ఉజాగరే, ఎస్., & టెగ్గెల్లి, పి. (2015). FPGA -ఆధారిత బారెల్ షిఫ్టర్ రూపకల్పన మరియు అమలు. *ఇంటర్నేషనల్ జర్నల్ ఆఫ్ అడ్వాన్స్డ్ రీసెర్చ్ ఇన్ కంప్యూటర్ ఇంజనీరింగ్ & టెక్నాలజీ (IJARCET), 4 (1)*, 101-104.

రెడ్డి, కెఎన్, & బాబు, ఎస్కె (2016). FPGA -ఆధారిత 32-బిట్ బారెల్ షిఫ్టర్ రూపకల్పన మరియు అమలు. *[ప్రచురణ వివరాలు లేవు]*.

DESIGN AND IMPLEMENTATION OF FPGA BASED LOW POWER 32- BIT BARREL SHIFTER

Gudapati Lakshmi Bhavani¹, M.Nithin², T.Lokesh^{3*} V Sheshank⁴

^{1,2,3,4}KG REDDY College Of Engineering & Technology, Hyderabad,Telangana-India

⁴Dept of Electronics and Communication Engineering

Corresponding Author *: lakshmisrinicas.2010@kgr.ac.in

Abstract-This project presents the design and implementation of a low-power, high-performance 32-bit barrel shifter on an FPGA. A barrel shifter is a combinational circuit widely used in modern processors to perform fast bitwise shifting and rotation in a single cycle, unlike conventional sequential shifters that require multiple cycles. The design, developed in Verilog HDL, was simulated and synthesized on FPGA platforms such as Xilinx. It supports four operations: logical left shift (LSL), logical right shift (LSR), arithmetic right shift (ASR), and rotate right (ROR). A hierarchical multiplexer structure is employed to minimize delay and optimize hardware utilization. Post-synthesis results confirm correct functionality, efficient performance, and low resource usage, consuming only a small fraction of Look-Up Tables (LUTs) and Input/Output Blocks (IOBs). Power analysis shows minimal static and dynamic power consumption, with most dynamic power attributed to I/O activity, making it well-suited for embedded and portable systems. Overall, the design demonstrates a scalable, low-power, and robust barrel shifter architecture ready for deployment in modern digital applications.

Keywords:

Barrel Shifter, FPGA Implementation, Low-Power Design, High-Performance Architecture, Verilog HDL, Logical Shift Left (LSL), Logical Shift Right (LSR), Arithmetic Shift Right (ASR), Rotate Right (ROR), Multiplexer-Based Design, Resource Utilization.

1. Introduction

A barrel shifter is a combinational circuit that performs fast bitwise shifting or rotation in a single cycle, making it widely used in processors, DSP, cryptography, and embedded systems. Unlike conventional shift registers, it employs a multi-stage selection using multiplexers (MUX) to achieve efficient operation. Shifting is vital for multiplication, division, floating-point normalization, and bit manipulation. FPGAs provide an ideal platform for implementing barrel shifters due to their parallelism and high-speed processing. A 32-bit barrel shifter can be designed using a hierarchical structure with 1, 2, 4, 8, and 16-bit stages, minimizing propagation delay and

optimizing resource utilization. The design is implemented using Verilog or VHDL and tested on FPGA platforms such as Xilinx Vivado/ISE or Intel Quartus. This project demonstrates the advantages of hardware-based shifting over software methods, highlighting improvements in speed, efficiency, and parallelism. The implementation further emphasizes the importance of barrel shifters in modern applications such as embedded systems, cryptography, and machine learning accelerators. The research objectives are:

1. To design and implement a low-power, high-performance 32-bit barrel shifter using Verilog HDL on an FPGA platform, ensuring support for multiple shift and rotate operations such as LSL, LSR, ASR, and ROR.
2. To optimize the barrel shifter architecture using a hierarchical multiplexer-based structure, with the goal of minimizing propagation delay, reducing hardware resource utilization, and achieving scalability for integration into larger digital systems.
3. To evaluate the functional correctness, resource efficiency, and power consumption of the proposed barrel shifter through FPGA synthesis and post-implementation analysis, validating its suitability for embedded and portable low-power applications

2. Literature Survey

Recent research in digital hardware design has focused on enhancing performance, reducing power, and improving scalability across diverse applications. Papageorgiou (2025) introduced novel scheduling and shifter network architectures for high-throughput signal and image processing. Sharma, Purohit, and Asati (2025) implemented a single-cycle LDPC encoder on FPGA for 5G systems, while Li et al. (2025) proposed a 14-bit hybrid SAR ADC with self-calibration for accuracy and low power. In cryptographic hardware, Lee et al. (2025) studied interleave sizes for the GIFT-128 cipher, and Rahul et al. (2025) designed a RISC core using Vedic mathematics to save power and area. Similarly, Izadi and Jamshidi (2025) presented a low-power approximate multiplier, and Sumarudin et al. (2025) developed a dual-mode accelerator for deep Q-networks in edge computing. Power-efficient design has also gained attention, with Joyce et al. (2025) optimizing power grids in congested datapaths, Agarwal, Garg, and Kumar (2025) advocating FPGA-based low-power systems, Mule and Mande (2025) reducing FFT complexity via CORDIC modules, Modia et al. (2025) implementing a Vedic multiplier, and Erna et al. (2025) realizing a high-throughput Huffman codec. Specific to barrel shifters, Devamane et al. (2015) emphasized functional correctness and resource efficiency, while Reddy and Babu (2016) highlighted the suitability of 32-bit FPGA-based barrel shifters for high-speed data processing. These works collectively provide the foundation for current research

on scalable, low-power, and high-performance barrel shifter architectures for modern embedded and portable systems.

3. Methodology

To achieve the objectives flowchart of the 32-bit barrel shifter represents the systematic steps taken during the entire design and implementation process. It begins with understanding the requirements of the barrel shifter, including its input/output structure and the types of operations it should perform. Once the problem is defined, the next step is to design the logic using Verilog HDL.



Figure1: Flowchart of the 32-bit barrel shifter.

The flowchart of the 32-bit barrel shifter represents the systematic steps taken during the entire design and implementation process. It begins with understanding the requirements of the barrel shifter, including its input/output structure and the types of operations it should perform. Once the problem is defined, the next step is to design the logic using Verilog HDL, specifying how the data will be shifted or rotated based on control signals.[3]

4. Result Analysis

The waveform shown in the behavioral simulation result represents the functional verification of a 32-bit barrel shifter using Verilog. The simulation is conducted to observe the output response of the barrel shifter for different shift modes and input conditions over time. The input D[31:0] carries the 32-bit data to be shifted, and S[4:0] specifies the shift amount. The control signal Mode[1:0] determines the type of shift operation: logical left shift (00), logical right shift (01),

arithmetic right shift (10), or rotate right (11). In the waveform, different time intervals (in nanoseconds) demonstrate behavioral simulation changes in these control signals and input data, with corresponding changes observed in the output Out[31:0]. For example, at one time instance, when the mode is set to "11" (Rotate Right) and the shift amount is "4", the output reflects the circular rotation of bits. The simulation confirms that the barrel shifter correctly interprets mode and shift inputs, producing expected output values in real-time. This validates the functionality of the design under different operating the models based on their performance metrics and analyze the results.

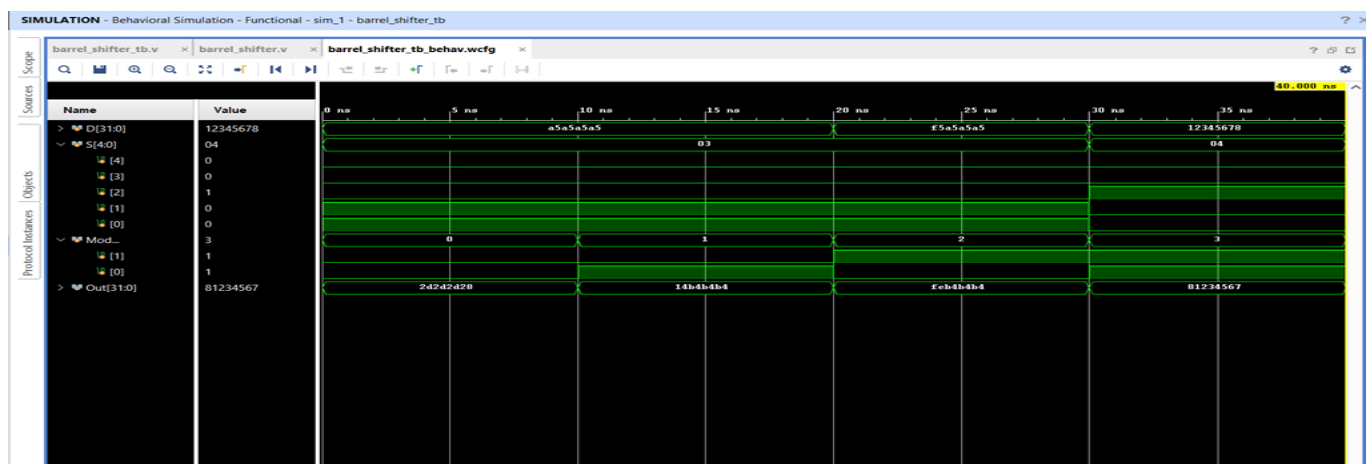


Figure2:Simulation for the behavioral simulation 32-bit barrel shifter using Verilog. The RTL (Register Transfer Level) schematic is a structural representation of HDL code generated after elaboration, showing how the design is logically organized into registers, multiplexers, gates, and control paths. For a 32-bit barrel shifter, the RTL schematic provides a high-level abstraction of the data flow, automatically generated from Verilog or VHDL. It includes input ports such as the 32-bit data line (D[31:0]), a 5-bit shift control (S[4:0]), and a 2-bit mode selector (Mode[1:0]). Internally, the design uses a hierarchical arrangement of multiplexers that shift data in stages of 1, 2, 4, 8, and 16 bits, enabling fast parallel shifting. The control logic, based on Mode[1:0], activates the required operation—Logical Left Shift (LSL), Logical Right Shift (LSR), Arithmetic Right Shift (ASR), or Rotate Right (ROR)—and the result is sent to the 32-bit output (Out[31:0]). The RTL schematic helps in verifying functional correctness, identifying issues such as unconnected ports or redundant logic, and documenting the intended architecture. In contrast, the post-synthesis schematic provides a gate-level representation of the same design after synthesis in tools like Xilinx Vivado, Synopsys Design

Compiler, or Intel Quartus. This schematic reflects the optimized hardware implementation mapped to FPGA resources, showing how the high-level RTL description is translated into standard logic gates and interconnections under constraints of area, speed, and power. For the 32-bit barrel shifter, this ensures that the design is functionally correct, resource-efficient, and performance-optimized before deployment.

Table1: REPORTS OF 8,16,32 BITS OF BARREL SHIFTER

LOGIC(8 bit)	USED	AVAILABLE	UATILIZATION
NO. of slices	18	860	1%
NO. of LUTs	33	1540	1%
NO. of IOBs	16	46	22%
LOGIC(16 bit)	USED	AVAILABLE	UATILIZATIN
NO. of slices	55	910	4%
NO. of LUTs	101	1590	4%
NO. of IOBs	22	56	46%
LOGIC(32 bit)	USED	AVAILABLE	UATILIZATOIN
NO. of LUTs	230	8000	3%
NO. of IOBs	71	100	71%

5.Conclusion

The 32-bit low-power barrel shifter was successfully designed and implemented, supporting logical, arithmetic, and rotational shift operations. Functional simulation verified that the design produced correct and predictable outputs under various control signals. Synthesis results showed minimal utilization of LUTs and I/O pins, making the design compact and easily integrable into larger digital systems. Power analysis confirmed low dynamic and static power consumption, with most dynamic power attributed to I/O activity in high-bit-width operations. The balanced distribution of logic elements helped reduce routing delays and improve overall speed.

Furthermore, the design demonstrated scalability across 8-bit, 16-bit, and 32-bit versions with minimal overhead. Overall, the proposed barrel shifter meets the essential constraints of power, performance, and area, making it highly suitable for modern VLSI systems and embedded applications.

6. References

1. Papageorgiou, N. (2025). Novel scheduling and shifter networks. In *Design and Architecture for Signal and Image Processing: 18th International Workshop, DASIP 2025, Barcelona, Spain, January 20–22, 2025, Proceedings* (Vol. 15569, p. 95). Springer Nature.
2. Sharma, A., Purohit, G., & Asati, A. R. (2025). FPGA-based implementation of single-cycle high-throughput LDPC encoder for 5G new radio. *IEEE Embedded Systems Letters*.
3. Li, S., Meng, Q., Zhang, L., & Wu, J. (2025). A 14-bit 2-MS/s hybrid-logic based SAR ADC with common-mode self-calibration. *Microelectronics Journal*, 1067.
4. Lee, M., Shin, H., Shin, M., Lee, D., Seo, S. C., & Hong, S. (2025). Best interleave size of GIFT-128 on 64-bit ARM processor. *Journal of Cryptographic Engineering*, 15(1), 1–16.
5. Rahul, D., Lakshmi, G. P., Reddy, G. M., & Ganorkar, A. M. (2025). Design and implementation of low power and low area RISC core using Vedic mathematics. In *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)* (Vol. 3, pp. 1–6). IEEE.
6. Izadi, A., & Jamshidi, V. (2025). LHTAM: Low-power and high-speed approximate multiplier for tiny inexact computing systems. *Computers & Electrical Engineering*, 123, 110215.
7. Sumarudin, A., Sutisna, N., Syafalni, I., Trilaksono, B. R., & Adiono, T. (2025). Low complexity and scalable architecture of dual-mode training and inference hardware accelerator for deep Q-network (DQN) based edge computing. *IEEE Access*.
8. Joyce, C. M., Upadhyay, P., Nishad, S., & Kakkar, A. (2025). PPA-aware power grid optimization techniques for congested high frequency datapath designs. In *2025 38th International Conference on VLSI Design and 2024 23rd International Conference on Embedded Systems (VLSID)* (pp. 308–313). IEEE.
9. Agarwal, P., Garg, T. K., & Kumar, A. (2025). Low-power embedded system design applications using FPGAs. In *Embedded Devices and Internet of Things* (pp. 1–20). CRC Press.
10. Mule, P. C., & Mande, S. S. (2025). FPGA-based reduction in complexity of FFT twiddle factor butterfly with embedded CORDIC module. *International Journal of Ad Hoc and Ubiquitous Computing*, 48(1), 34–45.
11. Modia, H., Chauhan, D., Patel, S., Gondalia, V., & Khistariya, J. (2025). Design and implementation of Vedic multiplier on Spartan-6 FPGA using Verilog HDL. *[Publication details missing]*.
12. Erna, G., Saidulu, V., Kudithi, T., & Banoth, A. N. (2025). FPGA implementation of high throughput encoder and decoder design of lossless canonical Huffman machine. *Results in Engineering*, 105037.
13. Devamane, S., Hanchate, A., Vagare, U., Ujagare, S., & Teggelli, P. (2015). Design and implementation of FPGA-based barrel shifter. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 4(1), 101–104.
14. Reddy, K. N., & Babu, S. K. (2016). Design and implementation of FPGA-based 32-bit barrel shifter. *[Publication details missing]*.